

# A SCORE-BASED INTERFACE FOR INTERACTIVE COMPUTER MUSIC

*David Litke and Keith Hamel*

School of Music  
University of British Columbia  
6361 Memorial Dr.  
Vancouver, B.C. CANADA V6T 1Z2  
*litke\_david@hotmail.com*  
*hamel@interchange.ubc.ca*

## ABSTRACT

The creation of interactive computer music often requires composers and performers to work within the paradigms of both instrumental music and responsive electroacoustics. The obstacles inherent in uniting these modes of operation have led the authors to develop an integrated working environment for interactive composition. NoteAbilityPro, a comprehensive music notation application, has been augmented to support real-time interaction with Max-MSP-Jitter or Pd. Control messages and notes with additional attributes can be embedded in a NoteAbilityPro score and sent to a network of computers running Max-MSP and/or Pd during an interactive performance. In order to allow the instrumentalist a degree of expressive freedom, score following using the `suivi.score` object in Max-MSP can be used to synchronize the score playback to the live performer. By integrating interactive performance controls in the notated score, a number of improvements have been realized, both in the working methods employed throughout the various processes of musical creation, as well as in the reliability and expressivity of the final performance.

## 1. INTRODUCTION

Musical works in which acoustic instruments and responsive electronics interact in a live performance pose a unique set of challenges to the creators, not only during performance but throughout the various stages of composition's development. In addressing the needs of musicians working in this genre, it can be useful to understand acts of musical creation not simply in terms of the insular categories of composition and performance, but rather as a dynamic interaction of a variety of processes. The act of composition will often involve various stages of improvisation and experimentation, responsive evaluation, structural organization, and notation. Similarly, a performer's activities can be described in terms of periodic phases of rehearsal and performance, involving multiple strategies for learning the music and developing an interpretation. Also important are archival activities, which aim to preserve a musical work for the purpose of future re-creation or revision. In the field of interactive computer music, the dynamic nature of the musicians' roles is particularly evident; composers and performers tend to be involved

in a wide range of activities throughout the development of a musical work, and the individual technical requirements for each work often demand unique treatments at each stage.

Throughout the processes of creation and realization of an interactive computer music composition, the composer must work both in the realm of live electronics and with acoustic instruments, activities which involve distinct creative tasks and which typically require different software applications. While the Max-MSP [10] and Pd [6] environments for real-time performance and interaction are well-suited to the generation and real-time control of electroacoustic sounds, their environments lack adequate score or time-based interfaces. Other music applications such as sequencers and digital audio editors, while equipped with highly developed time-based displays, do not offer sufficient capabilities for real-time interaction. In many instances, music notation software is used to notate the instrumental parts, but the interactive elements are not usually included in the score and most music notation programs are not designed to be functional during the performance of an interactive composition.

## 2. ENVIRONMENT

### 2.1 Overview

The environment presented here aims to bridge some of the difficulties inherent in developing, rehearsing, and performing interactive computer music. We have designed an integrated system that unites score-based and real-time electroacoustic paradigms. In order to develop such an environment, a number of enhancements were made to NoteAbilityPro [2] – a comprehensive music notation and editing application designed by one of the authors. In addition, Max-MSP and Pd modules were created to interact with NoteAbilityPro, enabling control messages and data to be passed between these applications.

### 2.2 NoteAbilityPro Extensions

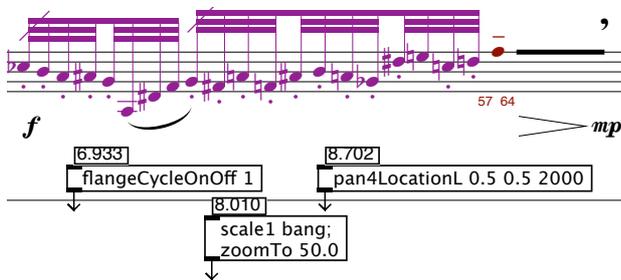
The extensions to NoteAbilityPro which support interactive computer music performance form the Integrated Interactive Music Performance Environment (IIMPE) [3]. This environment allows a NoteAbilityPro score to send control messages to 16 different IP addresses and ports during score playback, and 8 ports are available in NoteAbilityPro for receiving network

messages from the connected applications. Each computer in the network can be running Max-MSP-Jitter, Pd, or NoteAbilityPro. It is also possible for NoteAbilityPro to load Max-MSP or Pd patches remotely across the network, and to send the notated portions of the score to a `suivi.score` [7][8][9] object in order to facilitate score-following between a live performer and the NoteAbilityPro score.

## 2.3 Connections from NAPro to Max-MSP or Pd

### 2.3.1 Embedded Messages and Extended Notes

In a NoteAbilityPro score each staff is set up to direct its output either to standard sound destinations (MIDI, Apple DLS, Audio Units, etc.) or to a connected IP address and port. When the score is played, all connected applications receive score performance timing information (i.e. measure and beat data) as well as any embedded Max-MSP or Pd messages and extended notes notated on the staff. Max-MSP/Pd messages are entered into the score as text messages with the precise beat position indicated above the text. Each line of text ends with a semicolon and CR (following the messaging convention used in Max-MSP and Pd). Extended notes appear on the score in a different color and can include up to 16 additional parameters such as pitch-fraction, panning location, and *c2m* ratio. All player and synthesis modules of the UBC Max-MSP-Jitter Toolbox [4] can be controlled by the standardized data structure of extended notes. The figure below shows a score fragment containing both extended notes and Max-MSP/Pd messages.



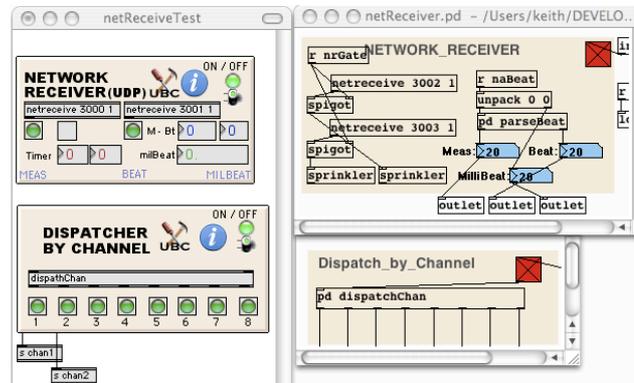
**Figure 1.** Extended notes and control messages embedded in a score.

Messages passed from NoteAbilityPro to Max-MSP or Pd are simply directed to corresponding receives in the patch. In the example above, the connected patch would include receive objects named `flangeCycleOnOff` (expecting a 0 or 1 message), `scale1` (expecting a bang) `zoomTo` (expecting a flonum) and `pan3LocationL` (expecting a list of two flonums and an integer ms duration). The data structure of extended notes includes channel information which determines where the data will be received in Max-MSP or Pd. In the example above all but the last of the extended notes is on channel 2 – the last note (indicated as 57 64) is on channel 1.

### 2.3.2 Max-MSP and Pd Objects

In order for Max-MSP and Pd to interact with NoteAbilityPro, messages are sent and received through `netsend` & `netreceive` [5] objects. In Max-MSP, modules

to facilitate sending and receiving messages are available as part of the UBC Max-MSP-Jitter Toolbox; similar patches have also been developed in Pd and are available from the authors.



**Figure 2.** Interfaces in Max-MSP and Pd for NoteAbilityPro messages and extended notes.

## 2.4 Connections from Max-MSP or Pd to NAPro

Enhancements to the playback controls of NoteAbilityPro enable it to respond to remote messages from Max-MSP, Pd or from another copy of NoteAbilityPro. Currently, the following messages and responses have been implemented:

Message	Data	Response
Tempo	flonum	Change current tempo
Pause	int (0–1)	Pause or resume playback
Start	bang	Start score playback
Stop	bang	Stop score playback
jumpToMeasure	int	Jump immediately to start of measure
sync	flonum	Adjust tempo to synchronize playback

**Table 1.** Performance messages responded to by NoteAbilityPro.

Of these messages, *sync* is used to synchronize a live performance with a NoteAbilityPro score (using score-following strategies), while *jumpToMeasure* can be used in improvisatory performances where a performer may want to treat the score in a more modular manner. The other messages are used as general playback controls; the ability to modify score playback remotely affords substantial flexibility over a wide variety of interactive performance configurations.

## 2.5 An Integrated Performance Setup

There are many different ways in which music and control data can be represented in a NoteAbilityPro score. In a simple configuration, a score can trigger Max-MSP events during playback somewhat like a multi-layered qlist. In a more complex configuration,

the NoteAbilityPro score can operate as a main control center, interacting with and responding to several applications running on multiple computers.

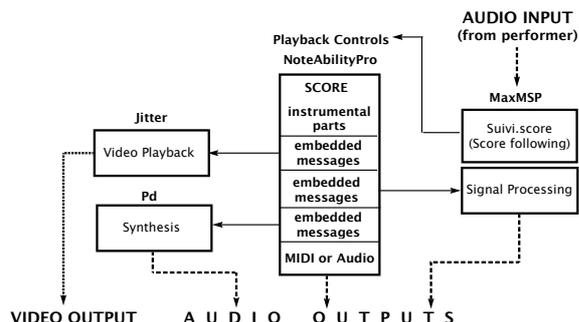


Figure 3. A complex interactive performance configuration.

### 3. SCORE-FOLLOWING STRATEGIES

#### 3.1 Score-following Implementation

A Max-MSP patch build around IRCAM's *suivi.score* object is used for score-following. Since the *suivi.score* object works most reliably in monophonic mode, a single instrument in the ensemble is selected to provide the audio signal to be followed during performance. The score data for that instrument is downloaded from NoteAbilityPro to a Max-MSP patch which formats the data so that it can be read by the *suivi.score* object. The encoded score includes all notes and rests except for any passages that have been explicitly marked for exclusion by the composer. A pitch-tracking module in Max-MSP determines the pitch and velocity of the live performance, and this data is passed to the *suivi.score* object which identifies where that event appears in the score. The tracking data is checked using the current score timing and the validity of previous assessments, and if deemed reliable, the current score data along with latency information is passed to NoteAbilityPro. Compensation for latency is necessary because the duration between the identification of the incoming pitch and retrieval of the score position depends on how quickly *suivi* can locate the event in its encoded score.

Within NoteAbilityPro there are four score-following modes; these modes control how closely the NoteAbilityPro score is synchronized to the incoming score data. Synchronization is achieved by adjusting the playback tempo either loosely (in order to maintain fluidity in the score playback) or more tightly (so that live performance and score are highly synchronized.) According to the kind of interactivity desired, different score-following modes are required. In scores containing autonomous musical passages whose rhythmic integrity must be retained, a looser score-following strategy is desired, while passages requiring highly responsive interactivity demand a tighter score-following strategy. All score-following parameters, including the mode, can be altered at any time during a performance.

### 3.2 Strategies

Over the past two years the IIMPE has been tested in a variety of interactive performance situations in order to develop reliable score-following strategies. In the most complex situations, such as when a performer is involved in free improvisation, score-following is ineffective since there is no representative score to follow. At the other end of the spectrum, where an instrumental part is represented using conventional notation (i.e. pitches and well-defined rhythms), score-following is easy to implement and extremely dependable. In scores where there is a mixture of conventional and non-standard notation (passages including extended instrumental techniques, microtones, improvisatory sections, or aleatoric passages) the score following must be enabled and disabled at various times during the performance. Since the NoteAbilityPro score can include control messages, and since patches can be built in Max-MSP or Pd to respond to those messages, various strategies have been developed for dealing with non-conventional scores. Score-following can be temporarily disabled and until being reactivated by a specific note, a certain period of time, or an external trigger such as a foot-pedal. During the development and simulation stages of an interactive composition an assessment of the effectiveness of score-following is made and additional strategies may be implemented according to the specific demands of the piece.

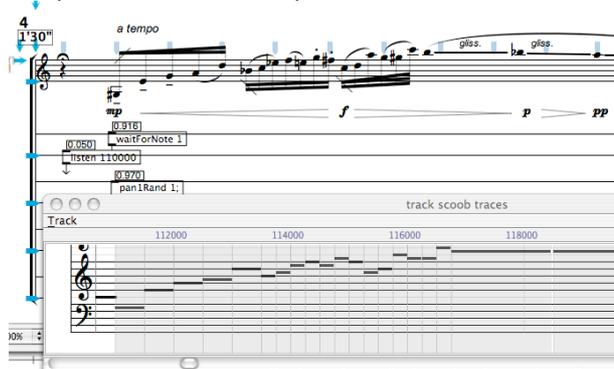


Figure 4. A NoteAbilityPro score and the corresponding passage as represented in the *suivi.score* display.

### 4. INTERACTIVE PERFORMANCE EXAMPLES

#### 4.1 Krishna's Flute

*Krishna's Flute* is a recent composition written by Keith Hamel for flute and interactive computers which uses the Integrated Interactive Music Performance Environment. The NoteAbilityPro score for *Krishna's Flute* contains one staff for the flute part (which is unmetred and represented in a rhythmically flexible manner), 7 staves of Max-MSP control messages (sent via ethernet to a second computer) and 2 staves containing extended notes (also sent remotely to the second computer). Score-following is performed using the techniques described above. All processing and synthesis events are sent from the score to Max-MSP during performance, and score-following (as well as manual override of the score-following) is used to ensure that all events in the score are kept closely synchronized with the live performance. For a performance of this work, the NoteAbilityPro score is played (with the flute part in the score muted) and all messages and embedded notes are sent to the connected

applications at the appropriate times. Latency between the performance tracking and the synchronization messages sent to NoteAbilityPro can be set before the performance and further adjusted during playback. However, latency has not proven to be a significant issue in any of the composition using this environment – even when wireless networking is used.

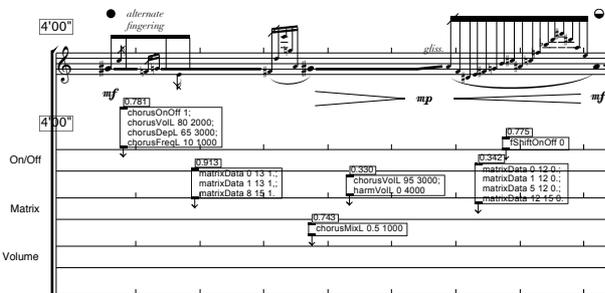


Figure 5. A score excerpt from Hamel's *Krishna's Flute*.

#### 4.2 Conduits

While the Max messages in the preceding example trigger discrete events or modify module settings, the following excerpt illustrates the environment's flexibility in controlling complex interactions. The passage shown in Fig. 6 is from the piece *Conduits* for clarinet and electronics by David Litke, and involves the creation of a complex texture by a sub-patch written in Max-MSP. Drawing from a reservoir of pitch and velocity data, this texture generator outputs a series of MIDI notes in random order, such that the probability of a particular pitch occurring at a given time is proportional to its velocity value. Other characteristics of the texture, such as the rate at which notes are played, the degree of regularity in the time between attacks, and the density (in terms of the amount of silence between attacks), are set via receives to the sub-patch.

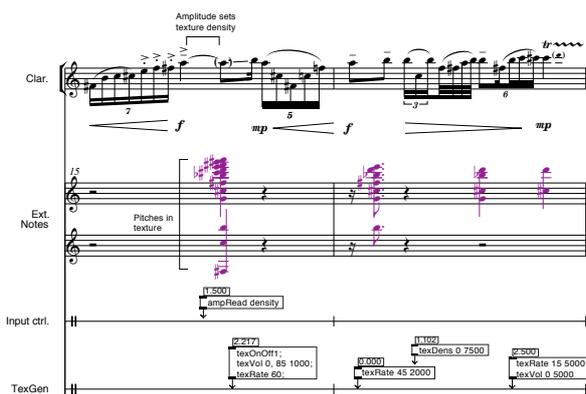


Figure 6. A score excerpt from Litke's *Conduits*.

In Fig. 6, the second and third staves contain a series of chords that have been entered as extended notes. During playback, these notes are not sounded as chords; rather, the pitch and velocity information associated with each note is sent to Max-MSP, providing the harmonic

materials to be used by the texture generator. Although in this example only pitch and velocity values are used, additional information could easily be attached to each of the extended notes if desired. As the passage progresses, new chords are sent to the patch, modifying the harmonic content of the texture. The other parameters of the texture generator are controlled by the Max-MPS messages situated on the fifth staff, which dynamically mould the broad contours of the texture.

The harmonic materials that are sent to the texture generator in this example were derived from spectral analyses of a gong sample, generated using tools developed in the programming language OpenMusic [1] and imported into NoteAbilityPro in MIDI format. While it would have been possible to store this data in Max and simply trigger its application with a score message, it is helpful to present a score representation of the pitch materials for a number of reasons. In developing the work, the composer is able to easily edit the chords, either to remove undesirable portions of the spectrum or to manually create synthetic harmonic fields; in crafting the instrumental portions of the work, the composer has a clear reference to the harmonic content of the spectrum; in addition, structural revisions to the work, such as measure additions or omissions, are greatly facilitated, since all of the messages that control the electronics are easily moved within the score. Furthermore, in learning and performing the work, the instrumentalist is provided with a visual indication of the activity and harmonic content of the electronics, aligning the score appearance with the aural experience.

As in *Krishna's Flute*, the *suivi.score* object is used to synchronize the playback of NoteAbilityPro with the clarinetist in *Conduits*. The changes in the texture generated in Fig. 5 are thus aligned with specific points in the clarinet part, regardless of tempo fluctuations during a particular performance. In this passage, the texture generator will shift to the harmony of the third chord when the performer arrives at the B on the third beat of the second measure, and the texture's volume control will begin a five-second decrescendo half a beat after that.

While interactivity between the performer and the computer has thus far been limited to maintaining synchronization despite tempo fluctuations, the example shown in Fig. 6 demonstrates another way in which the system can be responsive. Contained in the piece's Max-MSP patch is a sub-patch that samples the amplitude value of the signal coming from the instrumentalist's microphone when it receives a message to do so from NoteAbilityPro. This message also indicates how this value is to be used, sending the stored value to the appropriate receive object after scaling it within an appropriate range.

Referring again to Fig. 6, we can see that the message "density" will be sent to the *ampRead* receive in the amplitude capturing sub-patch half a beat after the arrival of the clarinet's high A. The amplitude value of that note is then scaled according to settings made in the sub-patch, the resultant value being sent to the texture generator's density input. The density of the texture

created during a particular performance will be proportionate to the intensity with which the performer plays the high A; this responsive interaction ensures that the dramatic balance between the clarinet and the electronic texture is maintained, and affords the performer a heightened degree of expressive freedom and control.

### **4.3 Structured Improvisation**

This environment can also be used in performances involving structured improvisation. Clarinetist François Houle has created improvisation modules where each page of the NoteAbilityPro score contains notated gestures as well as Max-MSP control messages and extended notes (which trigger samples and effects modules in Max-MSP). Using a foot pedal, François can direct messages to NoteAbilityPro in order to pause or resume score playback and to switch to any of the improvisation pages at any time. In performance, he is able to read the score fragments directly from the computer screen and has complete control over how slowly or quickly he proceeds through the improvisations. Score following is not used in these structured improvisations since the notated score is merely a rough map for the performer. The ability of the performer to navigate through a sequence of notated materials and to receive visual feedback on the status of all the electroacoustic components during performance has proven to be a significant aid to performers working in structured improvisatory situations.

## **5. CONCLUSIONS AND FUTURE WORK**

### **5.1 Assessment of the Environment**

In working with the IIMPE over the past couple of years, we have seen a number of significant advantages over antecedent methods of developing, rehearsing, and performing interactive computer music. By facilitating control over multiple components, the system can take advantage of the versatility and flexibility of a modular approach while retaining the intuitive working procedures of a centralized and integrated interface. The various components of a particular work, such as signal or video processing modules, score following systems, and specialized interactive patches, can be developed and operated separately (in different applications and possibly on different computers) while ultimately being coordinated by the master score. During experimentation phases, the composer is thus able to freely develop electroacoustic processes and materials in external applications such as Max-MSP, without being concerned about the ultimate timing of control messages. Subsequently, the composer can integrate controls for these materials into the score, being able to easily adjust their placement to discover desirable results and to organize materials in relation to other components of the music.

We also discovered that this environment allows the interactive composition to be simulated and tested easily and accurately, assisting the composer in evaluating the work during development. Synthetic instruments, either played directly from the NoteAbilityPro score or from a

recording of the instrumental part, can be used to test the balance, the score-following accuracy, the timing of events, and the effectiveness of the electroacoustic components. Changes are almost exclusively made in the score, either by altering numerical values in the Max text boxes or by adjusting the location of messages or notes in the score. As a result of the improved aural image afforded to the composer during composition, we found that rehearsal times were considerably reduced, since differences between the simulations undertaken without the performer being present and the live performance of the work were surprisingly minor. In addition, rehearsals using this system have been more efficient than with other interactive setups, since the score playback can simply be reset to any measure location for the repetition of a particular passage, and the electronic portions will follow automatically. Performers have also found the integration of the electronic controls and the instrumental notation useful, facilitating their comprehension of the piece and enhancing their ability to make interpretive decisions. We have also found that the interactive performances were far more reliable and controlled than in our previous experiences.

From an archival perspective as well, this environment has important advantages. The fact that so much of the composition is contained in the NoteAbilityPro score simplifies the process of archiving the complete work and of future editing and modifying of the composition. As well, the separation of a given piece's control messages from the sound production patches will facilitate the re-creation of the work in the future, using computer systems that may be incompatible with present technology; because the interactions between the live instrument and the electroacoustic components are made explicit in the score, the compositional intent is preserved and can thus be performed with analogous equipment.

### **5.2 Future Enhancements and Extensions**

Score-following strategies still need to be improved, both within Max-MSP and within NoteAbilityPro. The `suivi~.score` object being developed at IRCAM should help facilitate the score-following of instrumental parts that are not primarily pitch based. The score-following implementation in NoteAbilityPro should also be improved so that it can better respond to and anticipate the tempo fluctuations of the performer. We also intend to explore methods of embedding not only the pathnames, but all patches and associated files within the NoteAbilityPro score, so that a single document will contain all the necessary components of the composition. Finally, we need to continue testing the Integrated Interactive Music Performance Environment in a wide range of interactive situations in order to expose the limitations of the system and to further refine and enhance the environment.

## **6. ACKNOWLEDGEMENTS**

We would like to acknowledge the Social Science and Humanities Research Council (SSHRC) of Canada

which provided financial support for this research, and the Media and Graphics Interdisciplinary Centre (MAGIC) at UBC which provided research space and administrative support. Thanks also to Diemo Schwarz, Norbert Schnell, and Serge Lemouton who provide valuable insights and assistance with the suivi objects, and to Bob Pritchard who helped in the designing and developing of the UBC Max-MSP-Jitter Toolbox modules. The UBC Max-MSP-Jitter Toolbox can be downloaded for free from:

<http://www.opusonemusic.net/muset/toolbox.html>

NoteAbilityPro runs on Macintosh OS-X computers (OS version 10.3.9 or higher). The software is commercially available from Opus 1 Music Inc – a demo version of NoteAbilityPro is available from:

<http://www.opusonemusic.net/muset/toolbox.html>

The suivi.score object is available to members of the IRCAM Forum. The score-following interface used in conjunction with NoteAbilityPro is available from the authors.

## 7. REFERENCES

- [1] Assayag, G. and Agon, A. OpenMusic 4.9. Ircam - Centre Pompidou, 2005.
- [2] Hamel, K. NoteAbility: A Music Notation System That Combines Musical Intelligence with Graphical Flexibility. In *Proceedings of the International Computer Music Conference*, 1994.
- [3] Hamel, K. "Integrated Interactive Music Performance Environment", *Proceedings of the International Conference for New Interfaces for Musical Expression (NIME 2006)*, 2006.
- [4] Hamel, K, and Pritchard B. *UBC Max-MSP-Jitter Toolbox*. (Version 0.90) Retrieved January, 2007 from <http://www.opusonemusic.net/muset/toolbox.html>.
- [5] Mathes, O. netsend and netreceive. Retrieved January 2007 from <http://www.akustische-kunst.org/maxmsp/>.
- [6] Puckette, M. Pure Data. In *Proceedings of the International Computer Music Conference*, 1996.
- [7] Orio, N and Schwarz, D. Alignment of Monophonic and Polyphonic Music to a Score. *Proceedings of the International Computer Music Conference*, 2001.
- [8] Orio, N. Lemouton, S. Schwarz, D. and Schnell, N. Score Following: State of the Art and New Developments. In *Proceedings of the International Conference on New Interfaces for Musical Expression 2003*.
- [9] Schwarz, D., Orio, N., Schnell, N. Robust "Polyphonic Midi Score following with Hidden Markov Models", *Proceedings of the International Computer Music Conference*, 2001.
- [10] Zicarelli, D., *Max 4.2 Reference Manual*, Cycling '74, [www.cycling74.com](http://www.cycling74.com). 2003.